Analysis and applications of explainable AI methods

Marta González Mallo

Supervisor: Dario García-Gasulla

Director: Ulises Cortés

April 2020

Universidad Politécnica de Catalunya (UPC) - Facultad d'Informàtica de Barcelona

Universitat de Barcelona (UB) - Facultat de Matemàtiques

Universitat Rovira i Virgili (URV) - Escola Tècnica Superior d'Enginyeria

Contents

1 Introduction

Nowadays Neural Networks are being used more and more in real-world applications to deal with activities that involve images and videos. They can be an attractive option to automatize tasks that have been always done by humans, we can find an example in the case of the "BBVA selfie and go" tool [1]. In this tool an AI model is able to replicate the role of a self-service's waitress by taking a photo of the costumer and its tray of food and charging the correspondent amount in the user's credit card. The robustness of this system has been demonstrated by having processed over 10,000 payments, allowing hundreds of employees to enjoy their meals more comfortably, speeding up checkout times by up to five minutes. Reviewing other studies we can see that the power of Neural Networks is not limited to replicate human capabilities, but outperform them. We can see in [2] a study that shows that faces contain much more information about sexual orientation than can be perceived or interpreted by the human brain. Their Neural Network model trained on 35,326 facial images could correctly distinguish between homosexual and heterosexual men in 81% of cases, and in 71% of cases for women while human judges achieved much lower accuracy: 61% for men and 54% for women.

Due to the mentioned potential of Neural Networks we are experiencing a sudden jump between theoretical models focused on obtaining good accuracy with an academic point of view to real-world applications where the model may be the ultimately responsible for a particular task. Real life scenarios present challenges that are not faced in theoretical cases where the metrics of the models use to be the primary considerations.

If we explore some publications we can see that sometimes the accuracy of an algorithm is not enough to determine how trustful the algorithm is. As an example, in [3] we can see a model that classifies human faces as young or elder. After running an explainable algorithm the authors detected that the algorithm had learned that "happy faces" correspond to young people, while "neutral and sad faces" correspond to elder people. The model was focused on the face expression instead of detecting other more representative aspects of a person's age as the its hair color or wrinkles, so that model could be easily fooled in a real-world application. This issue is known as the model bias. High bias occurs when the model makes wrong assumptions during the learning phase, usually due to inconsistencies in the training set. When the models are applied in real-world applications there can appear serious issues if the model has a bias.

A real case where ethical issues were present in a AI model can be observed in the most used algorithm by the US hospitals. This algorithm assigned a risk for the users to be readmitted in the hospital, and there was found a racist bias on it: black people had to be sicker than white people before being referred for additional help. Only 17.7% of patients that the algorithm assigned to receive extra care were black. The researchers calculate that the proportion would be 46.5% if the algorithm was unbiased [4]. The Racial bias is a problem that we can find in other applications, for instance ProPublica proved that an algorithm used to estimate a criminal's risk of reoffend was highly biased against black people [5].

Another case in which the reliability of the models is critical is the quality control process for the automotive industry where having an automatic process that replaces a human to check the parts can lead to a large amount of saving for the company. But it is important to ensure that the algorithm is looking properly at the defects, because one defect skipped can lead to a dangerous vehicle. We can find many more scenarios, as medical diagnosis, legal systems, etc. where we want to know what the algorithm is looking at to determine a classification, and not just a number that quantifies the accuracy on a testing set.

Neural Networks are sub-symbolic models that learn non-linear combinations of parameters from thousands of samples, and that is difficult for humans to interpret. For this reason they are considered

black box models. The main motivation of this project is to analyze some of the methodologies proposed to give some light to the Neural Network predictions, trying to understand and improve those methods.

The objectives of this thesis are:

- Explore the most popular methods proposed to explain AI predictions on images.
- Analyze the details of some selected methods and understand the differences among them.
- Use the algorithms to explain predictions of a particular model and check its performance in a complex case.

The rest of this thesis is structured as follows: Section 2 contains an overview of some of the most popular methods proposed to obtain explanations of the predictions of Convolutional Neural Networks. Section 3 contains the technical details of the selected algorithms from Section 2 and in Section 4 we can find a discussion about their effects and performance in a simplified configuration. In Section 5 is presented the experimentation carried out to validate the findings of Section 4 and finally Section 6 will contain the conclusions of this thesis.

2 Related Work

Model explainability in deep learning is the area focused on understanding the predictions of an AI model at different levels, from the mathematical point of view to a more empiric perspective. In this project the explanations that will be analyzed are the ones that facilitate the interpretation of the predictions and could help a human user to detect a bias in a model. Nowadays this field is an active area of research and there are different strategies that can be adopted to generate explanations that help human users understand the attributes that a specific algorithm used to determine a prediction. This section will present some of the most popular explainability algorithms in the field of Convolutional Neural Networks (CNN).

The type of explanations studied in this project are those that aim to understand the label given to a specific image, so the whole model is explained through individual evaluations. There are two groups of algorithms to explain CNN's predictions that can be defined: external and internal methods. The first ones are those that generate explanations considering the model as a black box which can not be accessed. The second group of methods have access to the model architecture. External methods generate the explanations by modifying the inputs and observing the changes in the outputs, they are generally computationally expensive but also little dependent on the model, which makes them suitable for analyzing a wide variety of models. On the other hand, internal methods are defined specifically for each model, that makes them faster than external methods and also useful to help in the CNNs design and development. This project is focused on internal methods, although some external methods will be reviewed in this section.

2.1 External methods

One of the most known and used external method is LIME [6]. LIME stands for Local, Interpretable, Model-Agnostic, Explanations. This algorithm considers the model as a black box which can be a Neural Network, Support Vector Machine or any other Machine Learning model. In LIME, the information about the influence of the attributes of an input on its prediction is obtained by perturbing the input and observing the output.

LIME explains predictions on images by dividing them in several patches using segmentation to get meaningful pieces. Some patches are turned off randomly generating several samples of data points, where the Xs are binary vectors indicating which patches were turned on and the Ys are the output of the model evaluated for these samples. In Figure 1 we can see this process graphically represented, the Xs are the binary vectors under each sample and the Ys are the probabilities given to each vector, the reddest the border of the sample image the higher the probability assigned to it.

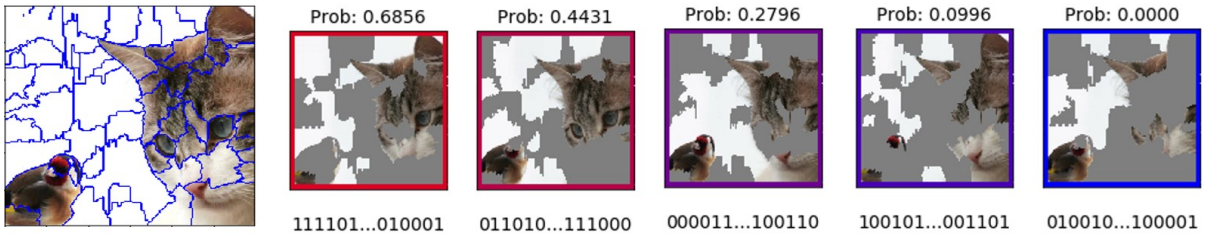


Figure 1: Fist step of LIME. The original image divided into patches at the left. At the right there are five generated samples and the probability given to them. The images have been generated based on the code provided by [6]

The second step is to do a Lasso Ridge Regression [7] between the binary vectors and the

probabilities, getting as a result a weight for the patches. The distance between the original data point and the samples is taken into account in the regression to get a local effect. Feature selection is used to obtain the most important patches.

The weight of the most important patches can be represented graphically using a color scale, in this case the authors in their code provided in [6] use green for the patches with positive influence on the class evaluated and red for the patches with negative influence. In the left image of Figure 2, we can see that for the Taby Cat class the cat's head and nose are the areas that contributes the most, while the Goldfinch bird is painted in red, which indicates a negative influence on the Taby Cat class. We evaluated the same image for the Goldfinch class and we can see in the right image of Figure 2 that the green area corresponds to the bird and the red patches fall on the cat.

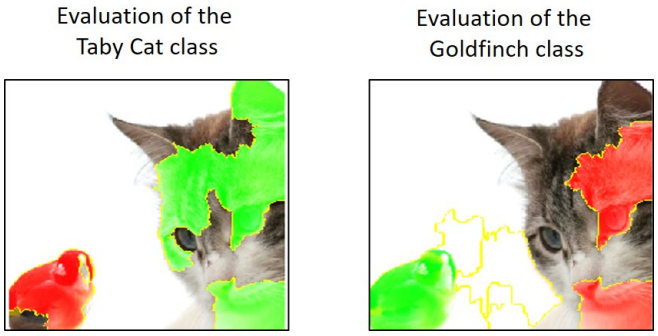


Figure 2: Left: Image evaluated for the Taby cat class, the positive patches in green are on the cat's head and mouth, the negative patches in red are on the bird. Right: Image evaluated for the class Goldfinch, the positive patches in green are on the bird, the negative patches in red are located on the cat. Images generated with the code provided by the authors in [6]

A desirable property of this algorithm is that it provides information about the influence of the different parts of the image in a certain class. For example, looking at the left image of Figure 2 we can make sure that the model knows that the lines on the cat's forehead make it a tabby cat. On the downside, it is a time consuming method, to get a robust explanation of an image it is required to generate at least 1000 random samples, it took 20 seconds to get a single image explanation on a Dell XPS 15 PC. Another minus point is that there is some stochasticity involved because it is not realistic to evaluate all the possible combinations of patches turned on and off and this generates a trade off between the time to get the evaluation and the robustness of the result. This second negative point is addressed in SHAP [8] that uses game theory methods to obtain consistency and local precision.

2.2 Internal methods

One of the most known and used internal method is LRP [9]. LRP stands for Layer-Wise Relevance Propagation. In the literature we can find this method used to explain Support Vector Machines and Recurrent Neural Networks but in this project it will be explored its performance explaining Convolutional Neural Networks.

As LIME, this method gives explanations for a particular image and a selected classification label. The explanations obtained with LRP are saliency maps were a value is assigned to each pixel of the image. The magnitude and sign of this value indicates how much the pixel influences the prediction and if the pixel promotes o inhibits the class prediction, respectively.

The process can be summarized in two steps. The first step consists on evaluating the image that we want to explain in a trained CNN, obtaining for each neuron of the network its activation

value in a forward pass through the CNN (x_i in Figure 3).

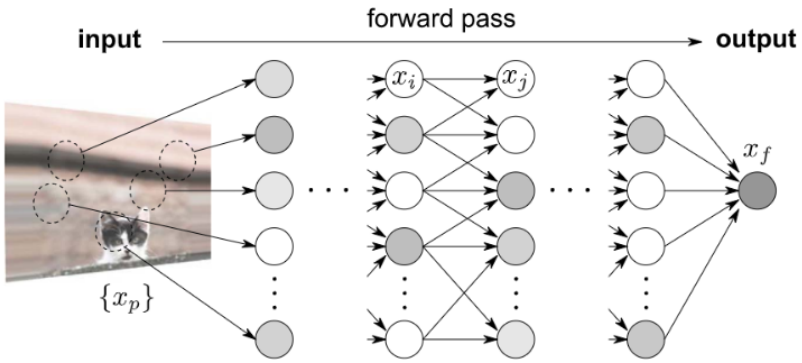


Figure 3: Representation of the first step of LRP. The activations of every neuron are computed using a trained model. Image from [10].

Once we have all the activations of the network we use them along with the model weights to propagate the relevance backwards, from the output neuron towards the input pixel representation (Figure 4). The output value of the class to evaluate just before the Softmax function is propagated through the network using a relevance rule reaching the input image. At the end of the process each neuron of the network has a relevance assigned that indicates the influence that the neuron has on the target class. The relevances that we are going to visualize are the ones of the input image's pixels.

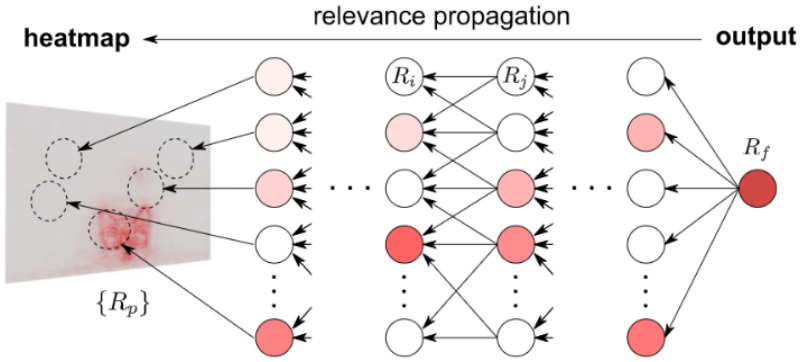


Figure 4: Representation of the second step of LRP. The relevances are propagated from the output neuron to the input image. Image from [10].

There are different rules proposed in the bibliography that we can use to propagate the relevance. In Section 3 we will present these rules and Section 4 is devoted to analyze its performance to understand the differences among them and which is the best option to get a good relevance propagation.

LRP consists on a forward and backward pass through a trained CNN so it is a faster method than the external methods presented previously like LIME or SHAP, where to obtain robust predictions it was necessary to generate and evaluate a large number of samples. Another desirable property of LRP is that the explanations generated are very detailed, as each pixel of the image is assigned a relevance value we can have a lot of information of which shapes of the image were the most relevant for the class. On the negative side, LRP is model dependant and it is not trivial to get an implementation of the algorithm to deal with large CNNs trained with every platform such as: TensorFlow, Keras, PyTorch, Ceffe, etc.

We can find other methods with a similar idea of LRP, like DeepLIFT [11] that decomposes the output prediction through layers by comparing the activation of each neuron to its reference activation or Guided Back Propagation [12] where the authors use partial derivatives applying the ReLU function together with the ReLU derivative across the network in the backward pass to obtain the explanations. For those architectures that have a Global Average Pooling (GAP) layer after the last convolutional layer instead of a dense layer we can use Class Activation Map (CAM) [13]. This algorithm uses the object-detector capability of the GAP layers to generate the saliency maps. In Figure 5 it is presented an overview of the process extracted from the paper [13] and we can see how it computes the saliency maps by scaling the GAP layers by their corresponding weight connection to the output neuron evaluated. We can find the improved versions of CAM: GradCam [14] and GradCam++ [15] which use gradients to provide coarse location maps and allow their use in many types of networks.

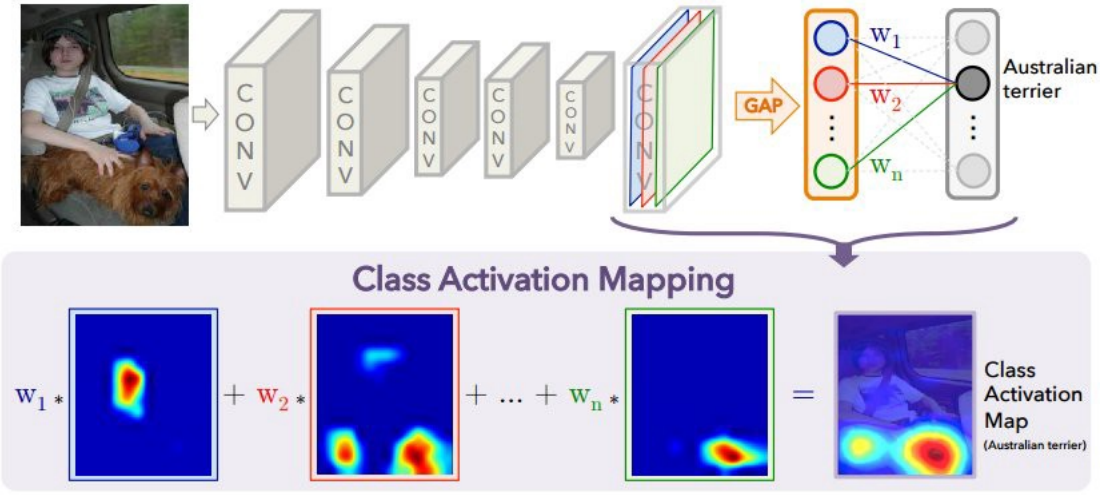


Figure 5: Class activation mapping: The predicted class score is mapped back to the previous convolutional layer to generate the class activation maps (CAMs). The CAM highlights the class-specific discriminative regions. Image from[13]

The robustness of some of these algorithms has been tried out in several studies. In [16] it is proposed to retrain the model to make it generate a new solution of weights and bias that maintain the accuracy but provides different saliency maps. The fine-tuning is done with a modified loss function that has two parts: the classical Softmax error and a term that quantifies the distortion of the new saliency map versus the original one, penalizing those saliency maps that are similar to the one generated by the original model. The authors propose different strategies of passive fooling (Location fooling, Top-k fooling, Center-mass fooling) and active fooling, and they end up proving that it is possible to fool LRP, Grad-CAM, and SimpleGrad using Adversarial manipulation. In [17] the authors prove that it is possible to distort the saliency maps by shifting the input images. This studies open a discussion of the reliability of this methods versus a hacker that wants to hide a bias in a trained model, showing that it is a possible option.

3 Theoretical review

In this section we will present the rules used in the LRP algorithm to propagate the relevance from the output layer to the input image through the CNN. The propagation process starts with the value prior to the Softmax function, which is called the logit, of the output neuron that corresponds to the class that we want to evaluate. This first value is propagated layer by layer using relevance rules. The nomenclature used to define the rules is:

- $W_{i,j}^{(l,l+1)}$ is the weight connecting the neuron *i* in layer *l* with the neuron *j* in layer *l*+1.
- $b_i^{(l)}$ is bias of neuron *i* in layer *l*.
- $m_j^{(l+1)}$ is the activation of the neuron *j* in the layer *l+1*, after the non-linearity function. In this project the only non-linearity used will be the ReLU function. Thus, the neuron activation is defined by the following function (1):

$$m_j^{(l+1)} = max(0, \sum_i m_i^{(l)} W_{i,j}^{(l,l+1)} + b_i^{(l)}) \tag{1}$$

- $R_{i\leftarrow j}^{(l,l+1)}$ is the relevance propagated from a neuron *j* in the layer *l*+1 to a neuron *i* in the layer *l*.
- $R_i^{(l)}$ is the total relevance of the neuron *i* in the layer *l*.

3.1 ϵ-LRP rule

The ϵ-LRP rule is the first rule proposed in [9] to propagate the relevance in LRP. In the ϵ-LRP (2) rule the relevance of neuron *j* is propagated to all the neurons connected to it from the previous layer. The relevance that a neuron *i* receives from *j* is proportional to the contribution that the neuron *i* gives to *j* i.e. $m_i{\cdot}W_{i,j}$. The $m_i{\cdot}W_{i,j}$ term takes into account the activation of *i* (m_i) and the strength of the connection between *i* and *j* ($W_{i,j}$). This contribution is normalized by all the contributions that *j* receives from the previous layer *l* $\sum_i m_i^{(l)} W_{i,j}^{(l,l+1)}$.

The ϵ acts as an stabilizer and avoids divisions by zero, but it absorbs part of the relevance.

$$R_{i\leftarrow j}^{(l,l+1)} = \left(\frac{m_i W_{i,j}}{\sum_i (m_i W_{i,j} + b_j) + \epsilon \cdot sign(\sum_i (m_i W_{i,j}))} \right) R_j^{(l+1)} \tag{2}$$

The total relevance of neuron *i* is the sum of all the relevances propagated to it from the neurons *j* of the upper layer *l+1* (3).

$$R_i^{(l)} = \sum_j (R_{i\leftarrow j}^{(l,l+1)}) \tag{3}$$

3.2 $\alpha\beta$-LRP rule

The $\alpha\beta$-LRP (4) was also presented in [9]. This rule separates the positive and negative contributions and assigns different weights (α and β) to each part. If we compare this rule with the ϵ-LRP (2) we can see that in the $\alpha\beta$-LRP case the normalization of the contributions is with respect of the neurons that have the same sign. In this case a neuron *i* that contributes positively to *j* will be normalized

just by the positive contributions that *j* receives, and the same for the case of the neurons with negative contributions. This rule has an interesting advantage with respect to the ϵ-LRP which is that it does not need the ϵ stabilizer.

$$R_{i\leftarrow j}^{(l,l+1)} = \left(\alpha \frac{(m_i^{(l)} W_{i,j}^{(l,l+1)})^+}{\sum_i (m_i^{(l)} W_{i,j}^{(l,l+1)})^+ + b_j^+} - \beta \frac{(m_i^{(l)} W_{i,j}^{(l,l+1)})^-}{\sum_i (m_i^{(l)} W_{i,j}^{(l,l+1)})^- + b_j^-} \right) R_j^{(l+1)} \tag{4}$$

The option recommended in [9] for the α and β parameters after empirical testing is α=2 and β=1.

3.3 Preset rule

The Preset option appears in the Innvestigate API [18]. It consists of a combination of the ϵ-LRP rule in the dense layers and $\alpha\beta$-LRP rule in the convolutional layers.

3.4 Bounded rule

The LRP Bounded rule (5) was proposed by [10] as a particular case of the Deep Taylor Decomposition to use in the first layer of CNNs, taking into account that the input in those architectures are images.

$$R_{i\leftarrow j}^{(l,l+1)} = \left(\frac{x_i W_{i,j} - l_i W_{i,j}^+ - h_i W_{i,j}^-}{\sum_i (x_i W_{i,j} - l_i W_{i,j}^+ - h_i W_{i,j}^-) + \epsilon} \right) R_j^{(l+1)} \tag{5}$$

l_i and h_i are the lower and the higher values of the input image respectively. X_i correspond to the pixels of the image.

If we look at (5) we can deduce that the relevance of the input pixels is scaled depending on the matching between the sign of its value x_i and the sign of its connection $W_{i,j}$ with the neuron *j* from the top layer: in the case of positive inputs if the input pixel and the weight have different sign the bounded rule modifies the input value to match the sign of the weight. If the input pixel and the weight have the same sign the sign will be maintained, but the magnitude will be decreased by l_i. If the input pixel is negative and the connection is also negative (i.e. the overall contribution $m_i \cdot W_{i,j}$ is positive) the absolute value of x_i will be increased by h_i, otherwise it will be decreased by l_i and the contribution will be negative.

3.5 Flat rule

The Flat rule (6) is another option available in the Innvestigate API [18]. In this case all the weights and input pixels are set to one in the formula (1). This is a drastic rule that ignores all the parameters of the layer and just propagates the relevances of the previous layer in a smoother way.

$$R_{i\leftarrow j}^{(l,l+1)} = \left(\frac{1}{\sum_i 1} \right) R_j^{(l+1)} \tag{6}$$

4 Analysis

This section is dedicated to analyzing the performance of the rules presented in Section 3. To be able to go to the details we decided to keep the CNN simple to allow the observation of weights, activations and relevances in different points across the network. The model analyzed has the following architecture:

Conv: 32 filters of 3x3

Conv: 64 filters of 3x3

MaxPoling: 2x2

Flatten

Dense: 512

Output: 10

The model has been trained using a batch size of 128 and 25 epochs.

We decided to keep the dataset as simple as possible to simplify the interpretation of the saliency maps and we used MNIST, which consists of grayscale images of handwritten numbers from 0 to 9.

In the literature there have been proposed several combinations and parameters for the LRP rules. In this project we are going to analyze different options using the Innvestigate API [18] which allows to combine rules and change parameters in a flexible way using Keras models.

In Figures 6 and 7 we can see an overview of the saliency maps obtained with the rules analyzed in this section. The color convention of the saliency maps is the following: red is used to indicate positive values in the saliency map and blue to indicate negative values. That means that the shapes colored in red are considered typical patterns of the class evaluated, while the areas colored in blue are considered negative evidences for the analyzed class. White pixels correspond to zero values in the saliency map, so we can interpret them as if they have a neutral effect on the analyzed class.

In Figure 6 we can see different saliency maps for an image of a handwritten number 3. Each column of the matrix corresponds to a different LRP rule and each row shows the class that has been evaluated. If we look at the class 2 case (third row in Figure 6) it is as if we asked the model: "Why this image of a handwritten 3 is/is not a 2?". In this row we can see some samples bordered in green and others bordered in red. Green samples indicate explanations we consider to be appropriate and particularly interesting, which for the number 3 evaluated for the class 2 are the ones where the top part of the number 3 has positive values (red pixels) and the rest has negative contribution (blue pixels), because just the top part of the number 3 is similar to a number 2. The samples bordered in red indicate what we consider to be wrong explanations and will be analyzed next. We can find another interesting case if we look at the explanations for the label 5 (sixth row in Figure 6): the bottom half of the number 3 is very similar to the bottom half of the number 5, so we can find this bottom half coloured in red indicating positive influence of these pixels in the label 5. In Figure 7 there is the same analysis for an image of a handwritten number 9. Here the most interesting instances can be found in the explanations of the label 4 (fifth row) bordered in green, where the top of the nine is colored in blue, indicating that those pixels are against the label 4, and actually the part colored in red seems a handwritten 4. A similar case can be found in the evaluation for the label 7 (eighth row).

The combination of rules analyzed in Figures 6 and 7 are:

- ϵ-LRP: ϵ-LRP rule applied in all the layers of the network.

- ϵ-LRP Bounded: Bounded rule applied in the relevance propagated from the first convolutional layer to the input image and ϵ-LRP rule applied in the rest of the layers of the network.
- ϵ-LRP Flat: Flat rule applied in the relevance propagated from the first convolutional layer to the input image and ϵ-LRP rule applied in the rest of the layers of the network.
- $\alpha 2\beta 1$-LRP: $\alpha 2\beta 1$-LRP rule applied in all the layers of the network.
- $\alpha 2\beta 1$-LRP Preset: $\alpha 2\beta 1$-LRP rule applied in the convolutional layers of the network and ϵ-LRP applied in the dense layers of the network.
- $\alpha 2\beta 1$-LRP Preset Bounded: Bounded rule applied in the relevance propagated from the first convolutional layer to the input image, $\alpha 2\beta 1$-LRP rule applied in the convolutional layers and ϵ-LRP applied in the dense layers of the network.
- $\alpha 2\beta 1$-LRP Preset Flat: Flat rule applied in the relevance propagated from the first convolutional layer to the input image, $\alpha 2\beta 1$-LRP rule applied in the convolutional layers and ϵ-LRP applied in the dense layers of the network.
- $\alpha 1\beta 0$-LRP: $\alpha 1\beta 0$-LRP rule applied in all the layers of the network.
- $\alpha 1\beta 0$-LRP Preset: $\alpha 1\beta 0$-LRP rule applied in the convolutional layers of the network and ϵ-LRP applied in the dense layers of the network.
- $\alpha 1\beta 0$-LRP Preset Bounded: Bounded rule applied in the relevance propagated from the first convolutional layer to the input image, $\alpha 1\beta 0$-LRP rule applied in the convolutional layers and ϵ-LRP applied in the dense layers of the network.
- $\alpha 1\beta 0$-LRP Preset Flat: Flat rule applied in the relevance propagated from the first convolutional layer to the input image, $\alpha 1\beta 0$-LRP rule applied in the convolutional layers and ϵ-LRP applied in the dense layers of the network.

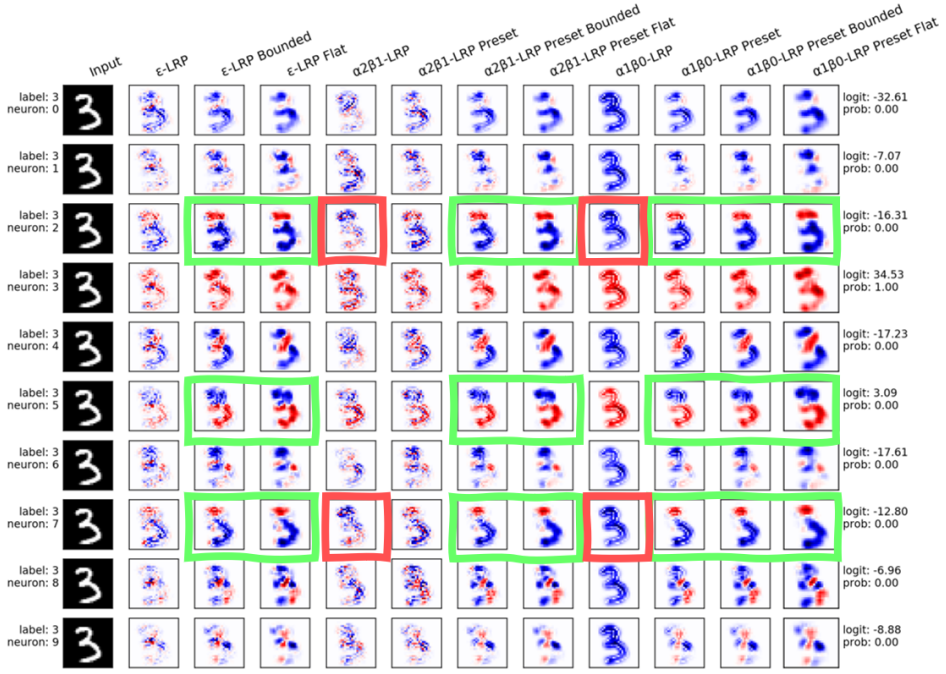


Figure 6: Overview of the 11 saliency maps obtained by evaluating a handwritten number three for the ten digits from 0 to 9. Green frames indicate particularly interesting explanations while red frames are considered by us wrong explanations.

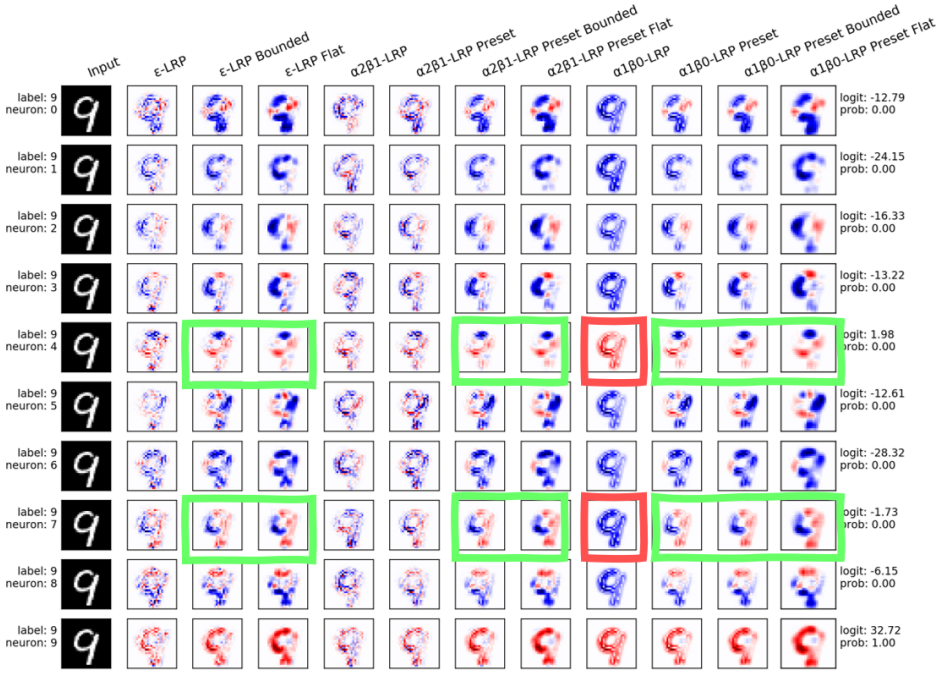


Figure 7: Overview of the 11 saliency maps obtained by evaluating a handwritten number nine for the ten digits from 0 to 9. Green frames indicate particularly interesting explanations while red frames are considered by us wrong explanations.

By observing Figures 6 and 7 we can get several insights. There are some combinations that provide noisier saliency maps than others, for instance those explanations obtained without the Bounded o Flat rules applied in the relevance propagated from the first convolutional layer to the input image representation. We can see other explanations where the saliency map has the inverted pattern of the main part of the methods, for example in the $\alpha 2\beta 1$-LRP case of third row of Figure 6, bordered in red we can see negative values on the top of the number three evaluated for the class 2, while the rest of the explanations and our intuition tell us that the top part of the number three is the most similar part to the number two. If we compare equivalent cases where the only differences are the α and β parameters we can observe different saliency maps but there is not a clear signal. We will propose three hypothesis that justify these effects observed.

4.1 Hypothesis 1: Dense layer option

In Figures 6 and 7 we have tested three LRP rules applied in the dense layers: ϵ-LRP, $\alpha 2\beta 1$-LRP and $\alpha 1\beta 0$-LRP. Just changing the LRP rule in the dense layers we have obtained very different results in the saliency maps, so the first hypothesis presented in this section will aim to justify those differences. To compare the effect of the three rules specifically on the dense layers we need to observe the relevance propagated in this part of the model. In Figure 8 we can see a representation of the model used and the part analyzed in this subsection is the colored in blue: the dense layer with 512 neurons and the MaxPool layer before the dense, which contains 64 channels with a size of 12x12 pixels each one. By observing just the final part of the network we will avoid getting confused by the interference with other rules across the network, because in Figures 6 and 7 we can detect signals in the choice of the rule that connects the first convolutional layer with the input image and in choice of the rule applied between the convolutional layers.

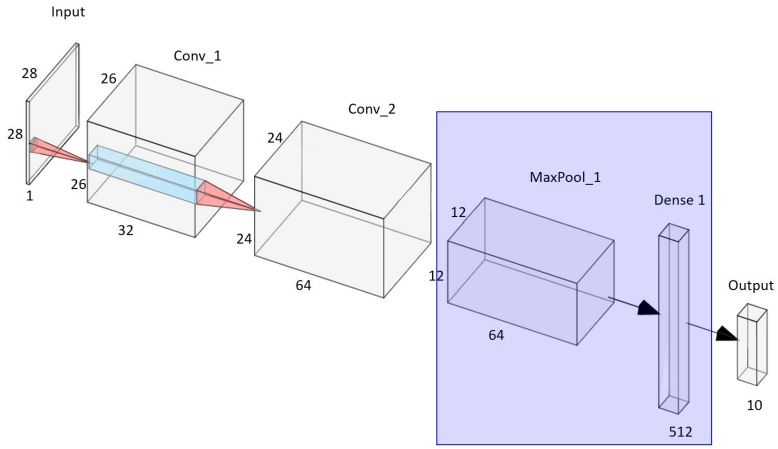


Figure 8: Representation of the model used, the blue area indicates the layers analyzed to discuss Hypothesis 1. Image generated using [19].

To discuss the effect of the different LRP rules on the dense layers we have selected the image of a handwritten number three evaluated for the class 2 because we saw an important signal in Figure 6 and it is simple to evaluate the correctness of its saliency map. The value of the output neuron of class 2 before the Softmax is -16.31, each rule propagates this value to the 512 neurons of the previous dense layer. In Figure 9 we can see the values of the relevances assigned to each one of the 512 neurons by each rule and we can observe that the ϵ-LRP rule has the opposite sign than the $\alpha 2\beta 1$-LRP rule and the $\alpha 1\beta 0$-LRP rule. If we take into account that the value propagated from the output class has a negative sign and we observe the rules, we can understand the reason of this behaviour:

- In the case of the ϵ-LRP its denominator is $\sum_i (m_i W_{i,j} + b_j) + \epsilon$ which is the same except for the ϵ as the value propagated from the output neuron. Both terms are almost compensated and the relevance propagated from the output neuron to the neurons of the prior dense layer is very similar to the real contribution of the neurons.

- In the case of the $\alpha 2\beta 1$-LRP and $\alpha 1\beta 0$-LRP the denominator of the rules do not compensate the sign of the output neuron, so this sign of the output value is propagated.

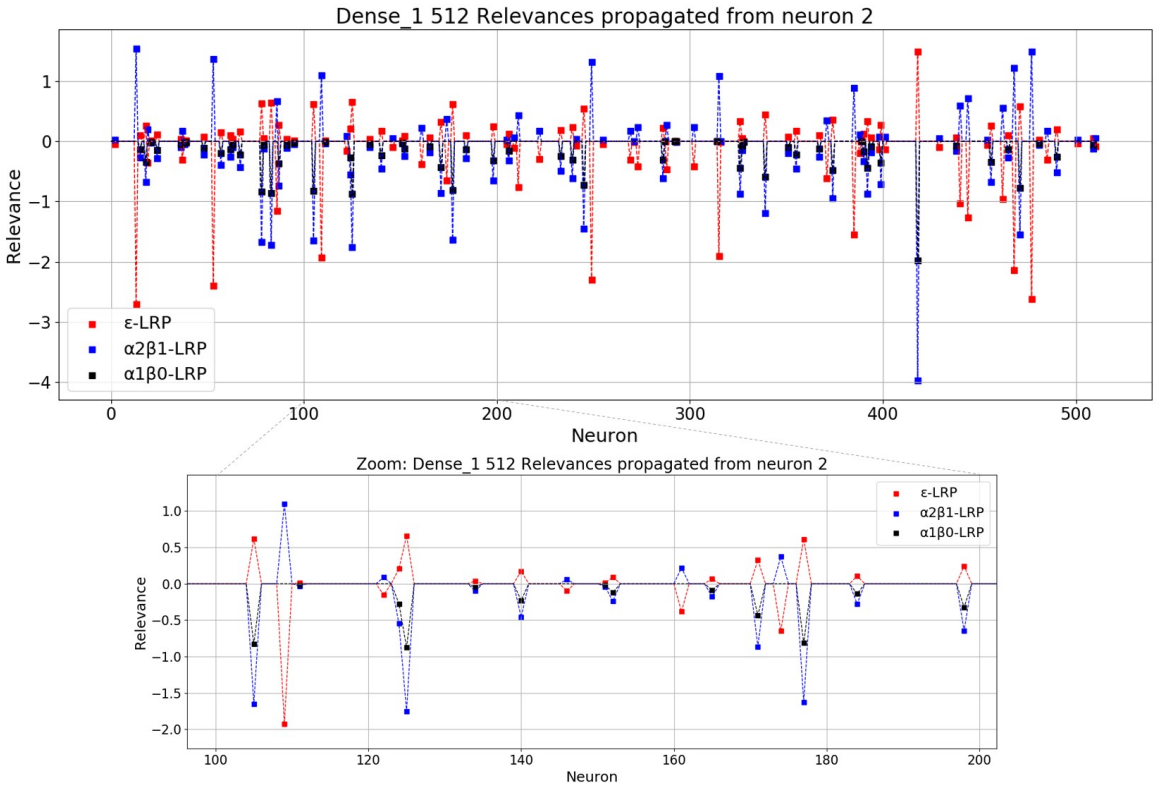


Figure 9: Comparison of the relevances assigned to the 512 neurons of the dense layer by the rules: ϵ-LRP, $\alpha 2\beta 1$-LRP and $\alpha 1\beta 0$-LRP. The image used is the handwritten number three evaluated for the class 2. The top image contains the values of the 512 neurons and the bottom image is a zoom of the values of 100 neurons.

Analyzing the relevance propagated by the three rules presented in Figure 9 we could deduce that the correct option is the ϵ-LRP case, because the relevance assigned to each neuron is almost the same as the contribution that this neuron has provided to the output class evaluated. The next point of the model where the rules are applied in a dense layers context is to propagate the relevance from the Dense 1 layer to the flatten vector of 9216 neurons. This 9216 neurons are reshaped in the MaxPool_1 layer that we can see in Figure 8. The MaxPool_1 layer contains 64 channels of 12x12 pixels each. In Figure 10 we can observe the sum of the relevances obtained in the 64 channels of this layer generated by the three rules:

- In the ϵ-LRP case the pattern obtained in this point of the network is a downscale of what we consider a good explanation of the image evaluated for the class 2: the top part of the shape could be a 2, so it is assigned positive relevances, but the rest of the number three is not similar to the class 2, so all the values are negative. If we look at Figure 6, the pattern obtained in this layer is less noisy than the saliency map obtained by many rules evaluated in the input layer.

- In the $\alpha 2\beta 1$-LRP case we can see the inverted pattern of the ϵ-LRP case, as we could expect after analyzing Figure 9. As the negative sign of the output class is being propagated the α term of the equation provides the negative relevances and the β term the positive ones. In this case we do not consider this explanation correct because it is indicating that the bottom part of the handwritten number 3 is similar to a handwritten number 2, which is not true.
- In the $\alpha 1\beta 0$-LRP case we see a saliency map where all the values are negative, again due to the propagation of the negative logit. The top area of the number three is the most negative part, this is because we are propagating just the positive contributions (α term of (4)) but with the sign inverted. We do not consider this explanation correct because it is indicating that the top part of the handwritten number 3 is less similar to a handwritten number 2 than the bottom part, which is not true.

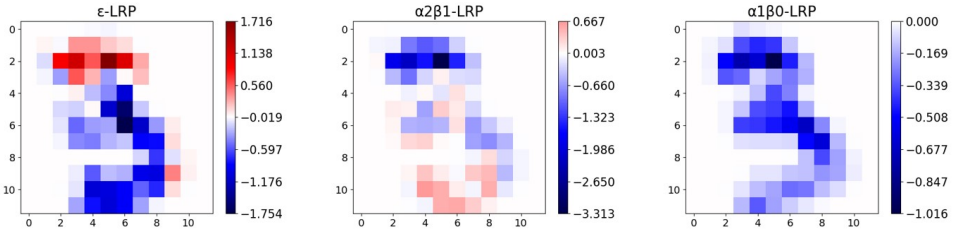


Figure 10: Comparison of the relevances propagated by ϵ-LRP, $\alpha 2\beta 1$-LRP and $\alpha 1\beta 0$-LRP rules to the MaxPool_1 layer. The image used is the handwritten number three evaluated for the class 2.

To observe the effect of the rules without the issue of the negative logit we can evaluate the same image for the class 5, whose output neuron has a logit of 3.09. In Figure 11 we can see that the values generated by the three rules in the Dense_1 layer have the same sign. When we observe in Figure 12 the saliency maps generated in the MaxPool_1 layer by the three rules we can see that the inversion of the maps due to negative logits is no longer an issue. The bottom part of the handwritten number 3 is the most similar to the number 5, and it is assigned positive relevance in the three cases. The saliency map generated by $\alpha 2\beta 1$-LRP contains some positive relevances in the top of the three that the ϵ-LRP does not, intuitively we would say that the ϵ-LRP is more correct because the top part of the number three is different than the top part of number five. The $\alpha 1\beta 0$ provides in this case a completely positive saliency map that is not helpful, because we could make the wrong assumption that the top part of the number three contributes positively to the class 5.

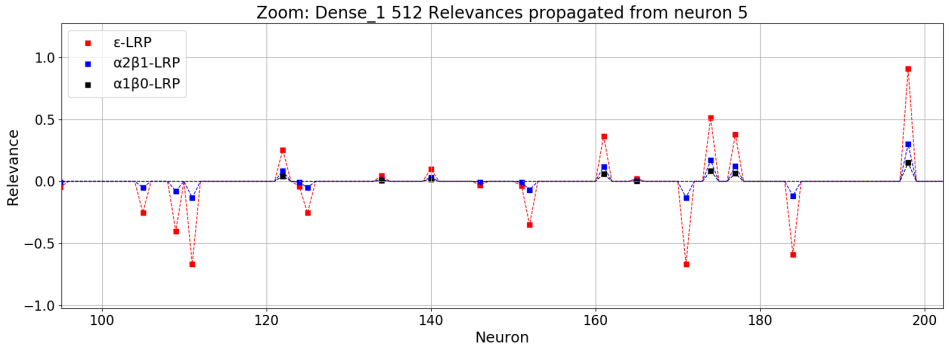


Figure 11: Relevances assigned to 100 neurons of the dense layer by the rules: ϵ-LRP, $\alpha 2\beta 1$-LRP and $\alpha 1\beta 0$-LRP. The image used is the handwritten number three evaluated for the class 5.

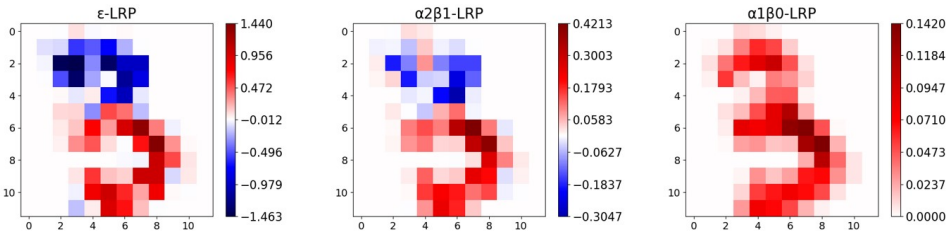


Figure 12: Comparison of the relevances propagated by ϵ-LRP, $\alpha 2\beta 1$-LRP and $\alpha 1\beta 0$-LRP rules to the MaxPool_1 layer. The image used is the handwritten number three evaluated for the class 5.

It is important to emphasize the difference between the relevance propagated from the output neuron and the relevance propagated from neurons inside the network. A negative sign in a neuron inside the network should be propagated, because this negative sign is indicating that this particular neuron is inhibiting the output neuron, but the sign of the output neuron should not be propagated. Positive contributions to a negative output still are positive contributions to the class.

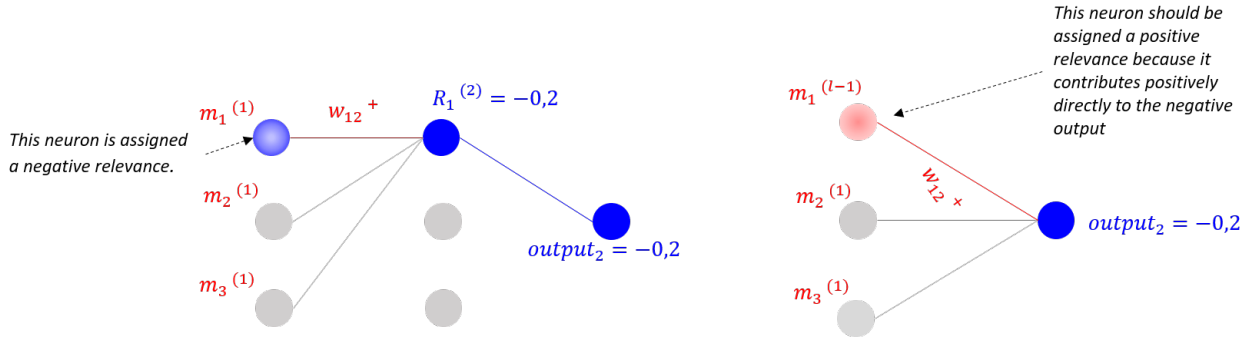


Figure 13: Schema clarifying the difference between the relevance propagated from an output neuron (right) and from a neuron inside the network (left).

The conclusions of Hypothesis 1 are:

- If we want to evaluate classes with negative logits we must use ϵ-LRP or other method to avoid the output sign propagation.
- $\alpha 2\beta 1$ can only be used in the dense layer prior to the output when the class analyzed has positive logits, otherwise the pattern gets inverted. When we used $\alpha 2\beta 1$ in a case with positive logits we did not perceive a clear advantage over ϵ-LRP in this simplified configuration.
- $\alpha 1\beta 0$ should not be used in the dense layer prior to the output neuron because just considering the positive contributions to the output class creates a pattern where all the pixels have the same sign and can lead to incorrect interpretations.

4.2 Hypothesis 2: Different behaviour of convolutional and dense layers

In Figures 6 and 7 we have tested three LRP rules applied in the convolutional layers inside the network: ϵ-LRP, $\alpha 2\beta 1$-LRP and $\alpha 1\beta 0$-LRP. Hypothesis 2 is devoted to understand which is the best option. After analyzing the Hypothesis 1 effect we consider that the best configuration to analyze Hypothesis 2 is to use ϵ-LRP in the dense layers and visualize the relevances of the Conv_1 layer. In Figure 14 we show in blue the part analyzed in this subsection. In the Conv_2 layer the relevances

are propagated from the MaxPooling connection, it is from the Conv_2 layer to the Conv_1 the first point in the network where LRP rules are used in a convolutional connection.

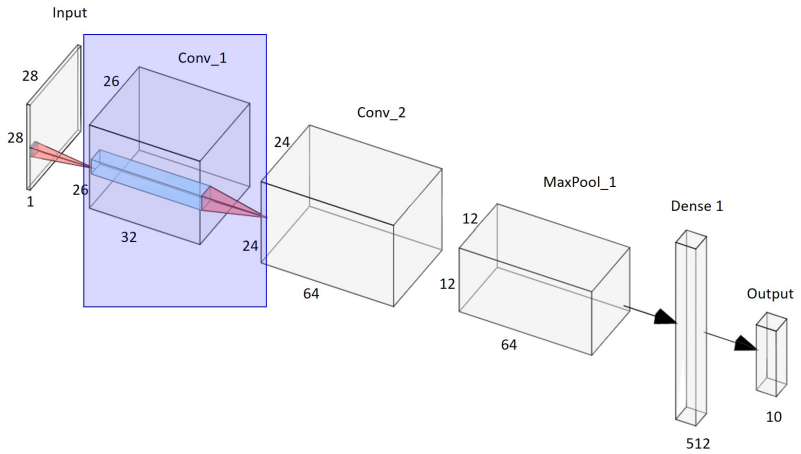


Figure 14: Representation of the model used, the blue area indicates the layer analyzed to discuss Hypothesis 2. Image generated using [19].

The rules analyzed to deduce which is the best LRP rule for the convolutional layers are:

- ϵ-LRP: in Figures 6 and 7 we can see that this option provides noisy saliency maps, but in Figure 10 and 12 we saw that the saliency maps obtained in the MaxPool_1 layer propagating relevances from the dense layers using this rule were the best. In this section we will understand the reason.
- $\alpha 2\beta 1$-LRP Preset: this option provides a better saliency map than ϵ-LRP, but worse than $\alpha 1\beta 0$-LRP Preset. Analyzing the relevances across the network we will try to justify this observation.
- $\alpha 1\beta 0$-LRP Preset: we will analyze this option because it provided a good result in Figures 6 and 7 and to visualize the propagation of positive contributions.
- $\alpha 1\beta 0$-LRP Preset: we will analyze this option to visualize the propagation of negative contributions.

Will analyze the relevance propagation in the case of the handwritten 3 evaluated for the class 2. As the Preset option implies that the dense layers use ϵ-LRP the relevances of the MaxPool1 are the ones presented in the left-most image of Figure 10 for all the rules analyzed in this subsection. In 4.1 we saw that the relevance on the convolutional layer prior to the dense layer already has the expected distribution of positive and negative relevances, even better than many of the saliency maps presented in Figures 6 and 7. In Figure 15 we can see the sum of the relevances of the 32 channels of Conv_1 layer propagated from the Conv_2 layer. We can see in the rigth-most image of Figure 15 that $\alpha 0\beta 1$-LRP inverts the pattern that is being propagated. There is no sense in having the top part of the 3 assigned a negative influence in the class 2 and the rest of the 3 positive influence, because the top part of the number 3 and 2 are similar and the rest is not. After looking at this saliency map we can understand why the $\alpha 2\beta 1$-LRP rule gives nosier maps than $\alpha 1\beta 0$-LRP: the α and β terms are being partially canceled, but not completely because the α term is twice the β term. In the ϵ-LRP case we can see a nosier pattern than $\alpha 1\beta 0$-LRP case, indicating that the ϵ-LRP is not the best option for the convolutional layer.

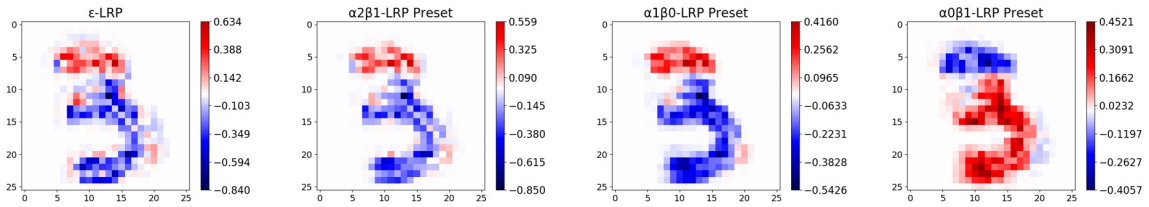


Figure 15: Comparison of ϵ-LRP, $\alpha 2\beta 1$-LRP Preset, $\alpha 1\beta 0$-LRP Preset and $\alpha 0\beta 1$-LRP Preset rules in the Conv_1 layer.

To force the limits of the conclusions presented in the previous paragraph we tested the following three rules:

- $\alpha 1\beta 1$-LRP Preset: if the positive and negative contributions are cancelled between them we expect this option to be nosy. This is what we see in the left most image of Figure 16.
- $\alpha 1\beta$(-1)-LRP Preset: if the positive and negative contributions are cancelled between them we expect this option to not be nosy, as we are avoiding the inhibition with the negative sign in β. This is what we see in the middle image of Figure 16.
- Flat Preset: we saw a correct saliency map in the convolutional layer prior to the dense layer. In this case we will propagate this relevance setting to one all the activations and weights of the convolutional layer. In the right most image of Figure 16 we can see that the map obtained is correct.

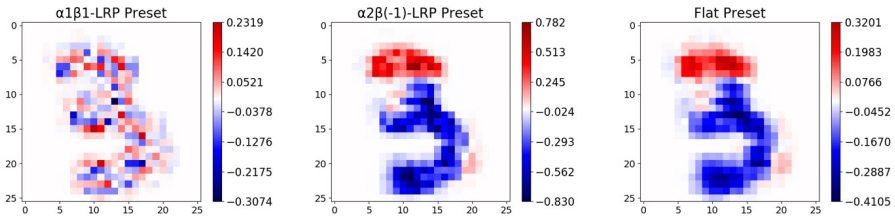


Figure 16: Comparison of the $\alpha 1\beta 1$-LRP Preset, $\alpha 1\beta(-1)$-LRP Preset and Flat Preset rules in the Conv_1 layer.

In Figure 17 we can see relevance propagation for the channel 16 of the Conv_1 layer. We can see the dependency between the $\alpha 1\beta 0$-LRP Preset relevance (middle image) and the activation of the channel (left image). The Flat rule propagates the same map to all the filters.

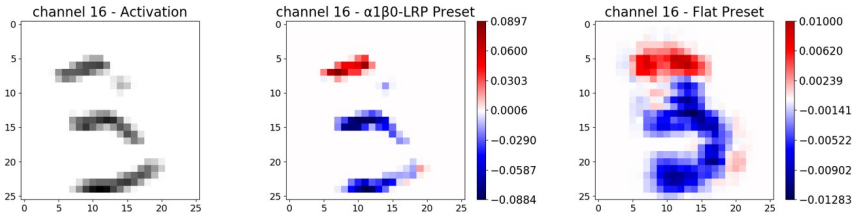


Figure 17: Comparison of the activation of the Channel 16 in the Conv_1 layer with the relevances assigned by the $\alpha 1\beta 0$-LRP rule and the Flat rule.

Comparing the insights obtained in Hypothesis 1 with the ones presented in this hypothesis we can see opposite signals: for the dense layers the best LRP rule was the ϵ-LRP, but for the convolutional layers using rules with negative signs invert the saliency map just after the dense layer generating noise in the propagation. We can assume that the convolutional layers act as features extractors, negative sign on their weights are used to generate edges on the activations and other representations, but a negative weight in convolutional layer is not related with negative influence on an output class, so it should not be propagated. The case of the dense layers is different, in that layers negative weight do indicate negative contributions to the class, so it makes sense to take their sign into account in the relevance propagation process.

The conclusions of Hypothesis 2 are:

- The ϵ-LRP is not a good choice to propagate the relevance in the convolutional layers.
- The $\alpha\beta$-LRP rule is a good option if the signs are not propagated. This can be achieved by using $\beta \leq 0$, so we avoid inverting the saliency map that comes from the convolutional layer prior to the dense layer which is already correct.
- If we use the Flat rule in the convolutional layer we are setting all the weights and activations to one and propagate the relevance conserving the pattern.

4.3 Hypothesis 3: First layer option

In Figures 6 and 7 we have tried five rules to propagate the relevance from the first convolutional layer to the input image: ϵ-LRP, $\alpha 2\beta 1$-LRP, $\alpha 1\beta 0$-LRP, Bounded and Flat. We can see that the options where the Flat and Bounded rules are applied in the first layer the saliency maps obtained are smoother. In this subsection we will analyze this effect by observing the relevances in the input layer, this area is indicated in blue in Figure 18.

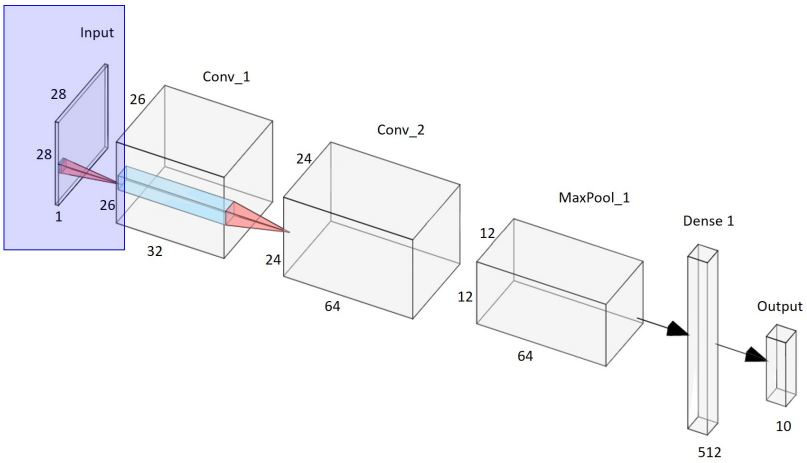


Figure 18: Representation of the model used with the input layer rule located. Image generated using [19].

To analyze this effect we decided to visualize the saliency maps of the handwritten image of a number three evaluated for the class 2. Following the conclusions of the two previous hypotheses we can discard the cases where ϵ-LRP and $\alpha 2\beta 1$-LRP is applied in the convolutional layers and all the non-Preset options, so we will analyze the $\alpha 1\beta 0$-LRP Preset, $\alpha 1\beta 0$-LRP Preset Bounded and $\alpha 1\beta 0$-LRP Preset Flat. We can see the comparison of the three saliency maps in Figure 19.

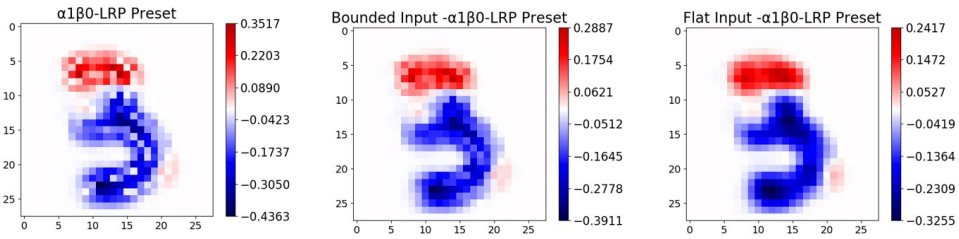


Figure 19: Comparison of the relevances in the input layer provided by the rules: $\alpha 1\beta 0$-LRP Preset, $\alpha 1\beta 0$-LRP Preset with Bounded input, $\alpha 1\beta 0$-LRP Preset with Flat input.

We saw in the Section 3 that the Bounded rule rewards those neurons that have a positive effect on the first convolutional layer. The Flat Rule ignores the value of the weights of the first layer and the inputs of the network, by setting all of them to one. In Figure 17 we saw the relation between the relevance map propagated to a layer and its activation when we use a LRP-based rule. We can see in Figure 19 that the Flat rule provides the smoothest saliency map. The activations of the first layer of convolutional networks tend to be coarser edges and the weights are simple patterns that can generate artifacts in the saliency maps, so using the Flat rule can be an interesting option. But with this simple configuration of small CNN and simple dataset we cannot assure that the combination of the $\alpha 1\beta 0$-LRP and Flat in the first layer is better than using the Flat rule in all the convolutional layers (Figure 20).

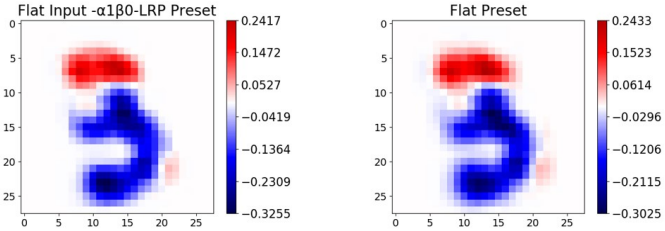


Figure 20: Comparison of the relevances in the input layer provided by the rules: $\alpha 1\beta 0$-LRP Preset with Flat input and Flat Preset. The image used is the handwritten number three evaluated for the class 2.

The conclusions of Hypothesis 3 is that is has been proved a reduction in the noise using the Flat rule in the input layer in this simple configuration.

5 Experimentation

In this Section we are going to explore the effect of the hypotheses presented in Section 4 with a more complex dataset and architecture. MNIST was a good option to get the firsts insights because the saliency maps are easy to interpretate, but its simplicity also limits the generalization of our hypotheses to more complex scenarios. The dataset selected for this section is Places 365 [20]. This dataset was created to be used for scene recognition as well as generic deep scene features for visual recognition. This dataset contains images of 365 categories of scenes. We can find images of places as airports, museums, gardens, discotheques, school classrooms, etc. In Figure 21 we can see three examples: casino, subway and greenhouse.

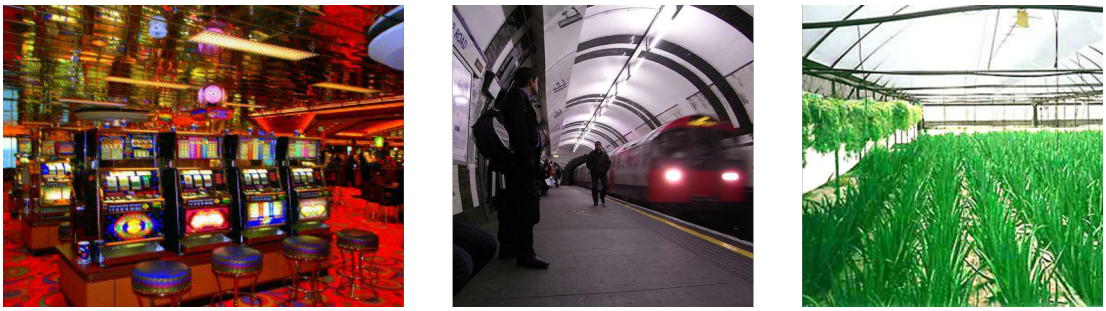


Figure 21: Example of three images of the Places365 dataset. Left: casino, middle: subway and right: greenhouse.

The dataset contains more than two million images divided as:

- Training set: 1,803,460 images, with between 3,068 and 5,000 per category
- Validation set: 36,500 images, with 100 images per category
- Test set: 328,500 images, with 900 images per category

We decided to use this dataset because the images of different places have many details and objects that can belong or not to a particular scene, so we can use LRP to see if the model is detecting those objects, and evaluate for different classes if the model considers that those objects belong to that class. With this option we can prove the generalization capability of the model beyond the accuracy of the testing set. The model is the one provided by the authors of [20] which has a VGG16 architecture and a 55.24% of top 1 accuracy and 84.91% top 5 accuracy. The original weights and model are deployed in Caffe, to use it with Innvestigate we have used the Keras version provided by [21].

In the next subsection we are going to validate the hypotheses presented in Section 4, next we will use the proposed LRP version to debug what the model has learned.

5.1 Hypothesis validation

The image used to validate the hypotheses is the recreation room that we can see in Figure 22. There are several objects in the image that are representative of that type of rooms, as the billiards and the ping pong table. We will evaluate the image for its maximum activated output neuron which is the correct class, Recreation room.

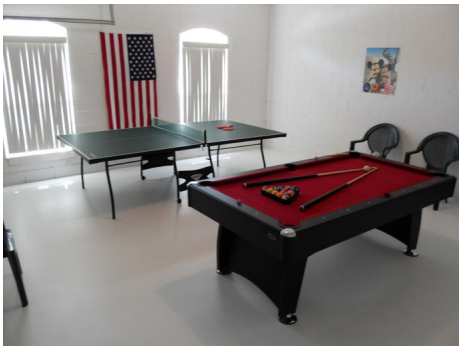


Figure 22: Recreation room.

5.1.1 Hypothesis 1: Dense layer option

In Hypothesis 1 we proposed to use the ϵ-LRP in the dense layers. To validate the hypotheses in the new context we will analyze the relevances in the max pooling layer previous to the dense layers for the rules: ϵ-LRP, $\alpha 2\beta 1$-LRP and $\alpha 1\beta 0$-LRP. In Figure 23 we can see that this layer is a MaxPooling of 512 channels of 7x7 pixels each one.

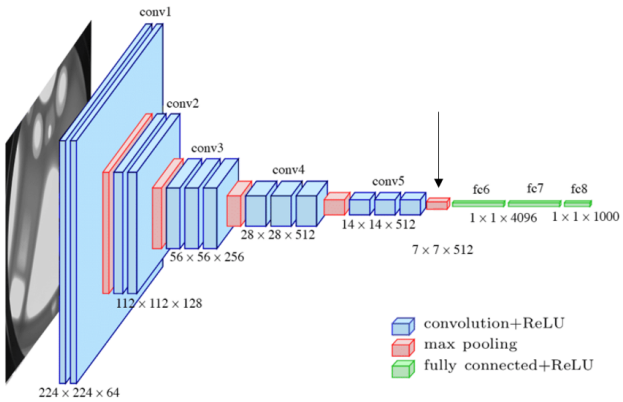


Figure 23: VGG16 architecture with the layer analyzed in this section indicated with a black arrow. Illustration from Max Ferguson.

In Figure 24 we can see the sum of the relevances of the 512 channels propagated by each rule. We can see that the relevance propagated by the ϵ-LRP contains negative values, while the relevance propagated by the $\alpha 2\beta 1$-LRP and the $\alpha 1\beta 0$-LRP rules provide completely positive maps.

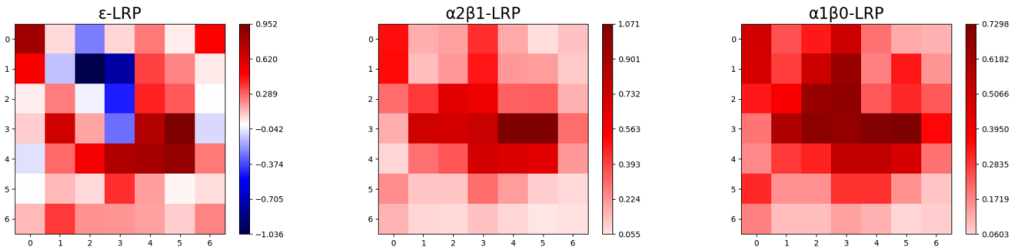


Figure 24: Relevances propagated by ϵ-LRP, $\alpha 2\beta 1$-LRP and $\alpha 1\beta 0$-LRP rules to the max pooling.

The interpretation of the relevances in the 7x7x512 max pooling layer displayed in Figure 24 is not easy due to the small size of the filters, so we have used the Flat rule to propagate the maps through the network to the input image to be able to compare them with the original image (Figure 25).

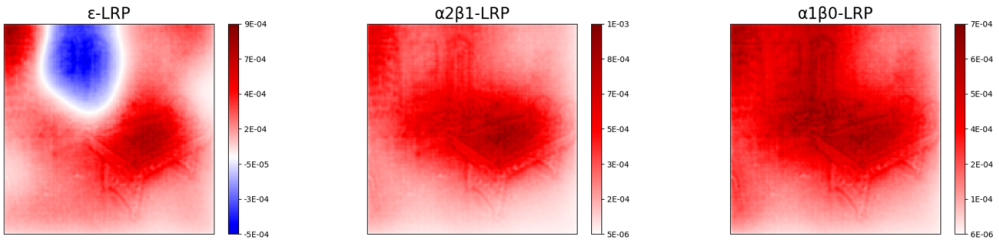
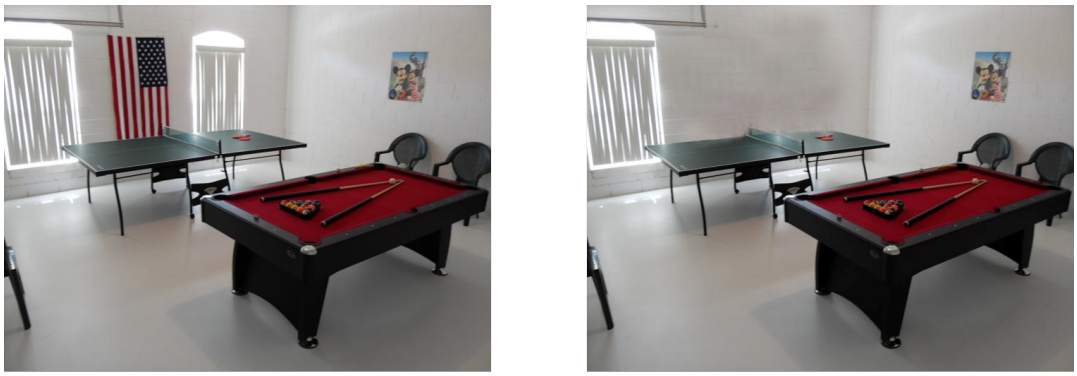


Figure 25: Relevances propagated using ϵ-LRP, $\alpha 2\beta 1$-LRP and $\alpha 1\beta 0$-LRP rules in the dense layers and the Flat rule in the rest.

In Figure 25 we can see that the ϵ-LRP indicates that the model considers that the USA flag of the wall and the window at its right are not typical objects of a recreation room. On the other hand, $\alpha 2\beta 1$ and $\alpha 1\beta 0$ indicate that these two objects have a positive influence on the class. In Section 4.1 we discarded the use of $\alpha 1\beta 0$-LRP in the dense layers because it ignores the negative contributions and generates incorrect maps. To check if this flag and window really were reducing the score of the output class as the ϵ-LRP rule indicates we removed them in the original image and we evaluated this modified image. We could see that the probability for the Recreation room class class increased when we remove the USA flag and the right window. In the left image of Figure 26 we show the original image, which obtained a probability of 0.9971 for the class Recreation room and at the right we show the modified image without the USA flag and the right window which obtained a slightly higher probability of 0.9995, maybe because in the training set there are no recreation rooms with flags on the walls. This confirms the ϵ-LRP saliency map indications and thus our Hypothesis 1.



Prob: 9.971E-01 Prob: 9.995E-01

Figure 26: Original image (left) and modified image without the flag on the wall and the right window (right).

5.1.2 Hypothesis 2: Different behaviour of convolutional layers

In the Hypothesis 2 we rejected the use of the ϵ-LRP rule in the convolutional layers and also using negative signs in the $\alpha\beta$ rule. To verify this hypothesis in this new context we have evaluated the

relevance propagated by the ϵ-LRP, $\alpha 1\beta 0$-LRP, $\alpha 0\beta 1$-LRP, $\alpha 1\beta$(-1)-LRP, $\alpha 2\beta 1$-LRP and Flat rules through the convolutional layer, using the ϵ-LRP rule in the dense layers in all the cases. In Figure 27 we can see the saliency maps obtained.

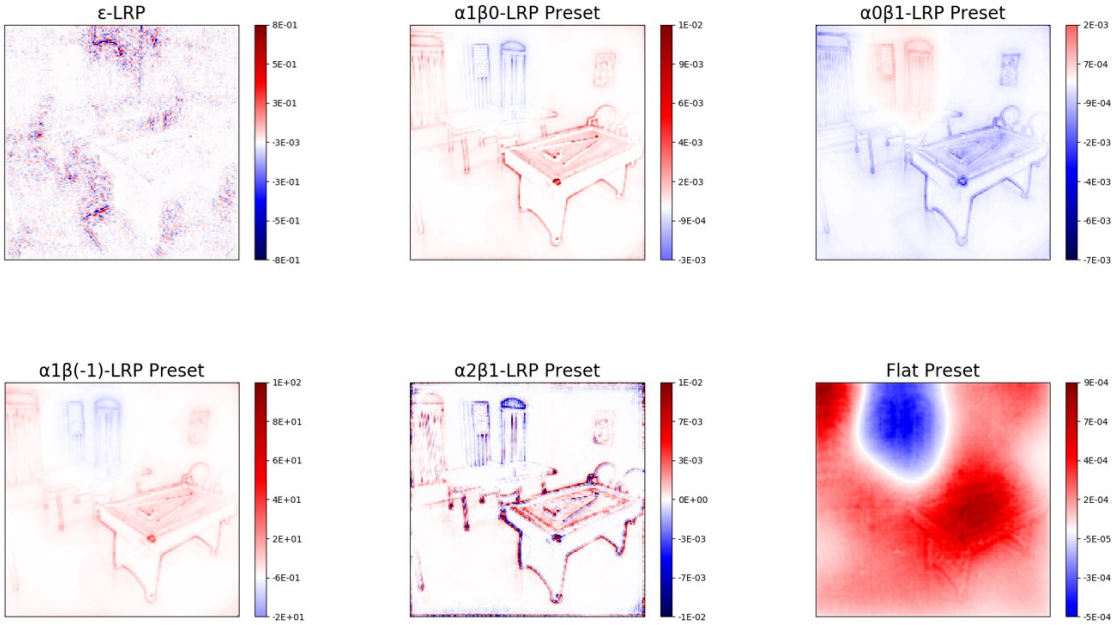


Figure 27: Saliency maps for the recreation room image obtained evaluated for its maximum activated class (Recreation room). Rules: ϵ-LRP Preset, $\alpha 1\beta 0$-LRP Preset, $\alpha 0\beta 1$-LRP Preset, $\alpha 1\beta$(-1)-LRP Preset, $\alpha 2\beta 1$-LRP Preset and Flat Preset. From Top to bottom and left to right.

If we look at the ϵ-LRP rule (in the top row - left image of Figure 27) we can see that the saliency map is not interpretable, there are positive and negative contributions placed randomly in the edges of the objects. ϵ-LRP Preset rule seems not to be an appropriate option to propagate the relevance through the convolutional layers.

The $\alpha 0\beta 1$-LRP Preset rule can be observed in the right image of the top row of Figure 27. In this case we can see the inverted pattern than the $\alpha 1\beta 0$-LRP Preset rule. As we saw in Section 4.2 the negative term of equation (4) inverts the pattern in the odd layers and it is the cause of the noisy map that we can see in the $\alpha 2\beta 1$-LRP Preset rule saliency map (Figure 27 bottom row middle image).

If we use the $\alpha 1\beta$(-1)-LRP Preset rule we can see in Figure 27 bottom row left image that the saliency map is smooth and correct, this is another confirmation of the bad performance that provides using negative term in the $\alpha\beta$ rule (4).

Finally, in the bottom row right most image of Figure 27 we can see the result of using the Flat rule in all the convolutional layers. As we saw in Section 4.1, the first convolutional layer after the dense layers has the correct distribution of positive and negative contributions but in a smaller scale: 12x12 in the case of the VGG16. Using the Flat rule we increase the size of this first filter without altering its distribution. Using other rules we get the edges of the images colored. With the MNIST dataset we got similar results in both cases, but in this configuration we see very different patterns. We think that both options are interesting and it may depend on the dataset which one is preferred. The observations of Hypothesis 2 are also confirmed in this context.

5.1.3 Hypothesis 3: First layer option

In Section 4.3 we saw some improvement when we used the Flat rule in the first layer. In Figure 28 we can see for the Recreation room image this effect with the $\alpha 1\beta 0$-LRP Preset rule and $\alpha 1\beta$(-1)-LRP Preset rule, and we can see that the combination of the edges with the colored surfaces provided by the Flat rule in the input generate saliency maps with more contrast.

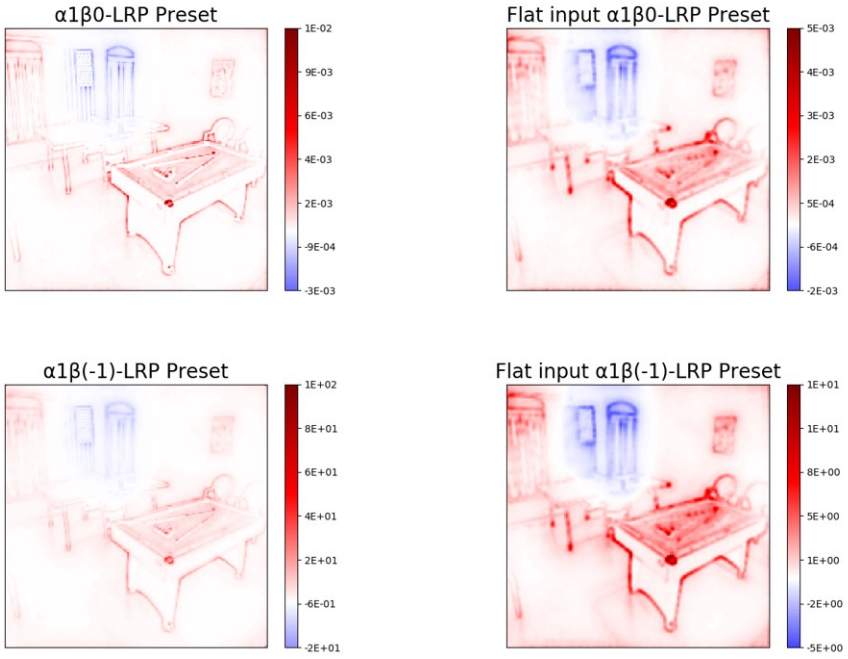


Figure 28: Top row: comparison of $\alpha 1\beta 0$-LRP Preset rule without the Flat rule in the input (left) vs the same rules with the Flat rule in the input (right). Bottom row: comparison of $\alpha 1\beta$(-1)-LRP Preset rule without the Flat rule in the input (left) vs the same rules with the Flat rule in the input (right).

5.2 Proposed LRP configuration

In this section we are going to explore the capability of LRP saliency maps to do model debugging in a complex dataset. In Section 4 we could interpret the saliency maps easily in the MNIST dataset, in this case we are going to use several images from the Places 365 dataset and see how interpretable the saliency maps are. For this task we will use a LRP version that satisfies the hypotheses presented in this thesis, there are two options:

- ϵ-LRP rule in the dense layers, $\alpha 1\beta 0$-LRP rule in the convolutional layers and Flat rule in the input layer.
- ϵ-LRP rule in the dense layers, $\alpha 1\beta(-1)$-LRP rule in the convolutional layers and Flat rule in the input layer.

Both combinations have demonstrated good performance in the two datatsets analyzed, but we decided to use the second one because it provided slightly higher contrast in Figure 28 and we have not found the $\alpha 1\beta(-1)$-LRP rule in the literature.

The first image that we are going to analyse is the Recreation room analyzed in the Section 5.1. If we ask the model "Why this image is/isn't a home office?" we can see in Figure 29 right most image the saliency map. The cue and balls of the billiard are detected as objects that do not belong to a home office, also the USA flag and the table-tennis net. On the other hand, the table structures, the left window and the frame on the wall are considered correct objects for the home office class.

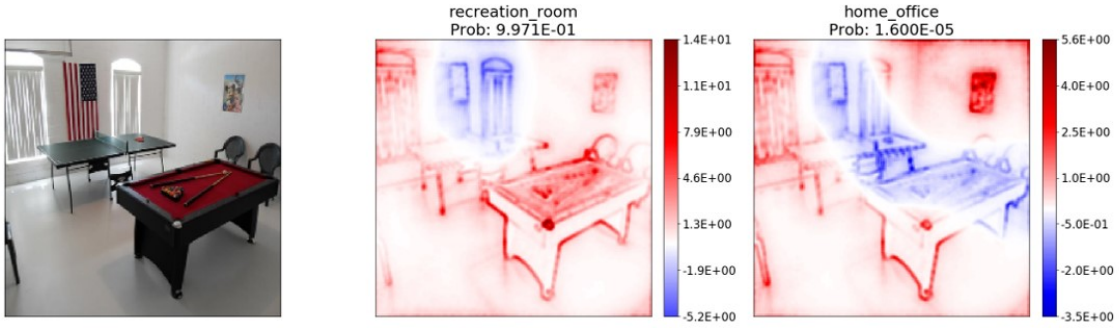


Figure 29: Recreation room evaluated for the maximum activation class (Recreation room) and for the class: home office.

In Figure 30 we can see the saliency map for an aqueduct image. In the middle image we see the saliency map for the most activated class, in this case is the correct label: aqueduct, and the main part of the image is considered correct for the class. If we look at the image evaluated for the class amphitheatre we can observe how the model is able to detect that the water on the ground is not something typical of amphitheatres, but it is the Romanic structure with several floors of windows.

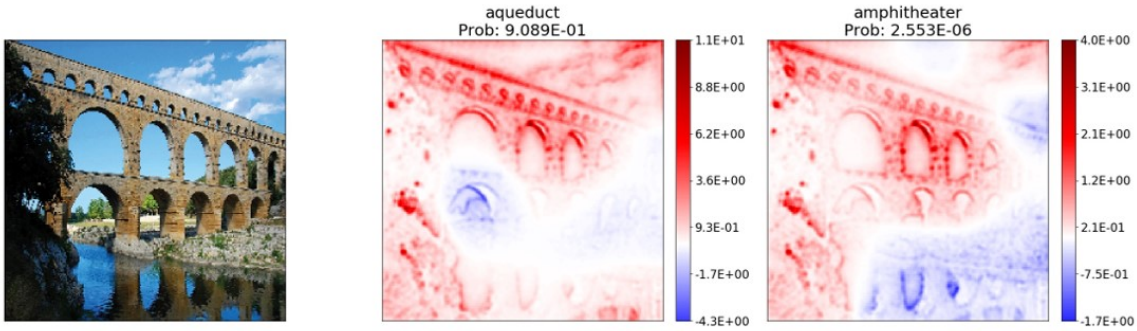


Figure 30: Aqueduct evaluated for the maximum activation class (aqueduct) and for the class: amphitheater.

In Figure 31 we can see the saliency map for an image of a patio. In the middle image we can see that for the maximum activated class the chimney is contributing negatively to the class, but the rest of the image has high positive scores. If we evaluate the image for the class bar (Figure 31 right image) we can see that the chairs and table are correctly identified and colored in red, but the model is also identifying that they are placed outside, because all the rest is coloured in blue.

In Figure 32 we can see some kind of art gallery. The model predicts that this scene is an artist's loft, we can see in the middle image of Figure 32 its saliency map. If we evaluate the image for the museum indoor class we can see in the right image of Figure 32 that the frames are considered typical objects of a museum, but not the objects on the ground. If we evaluate the same image

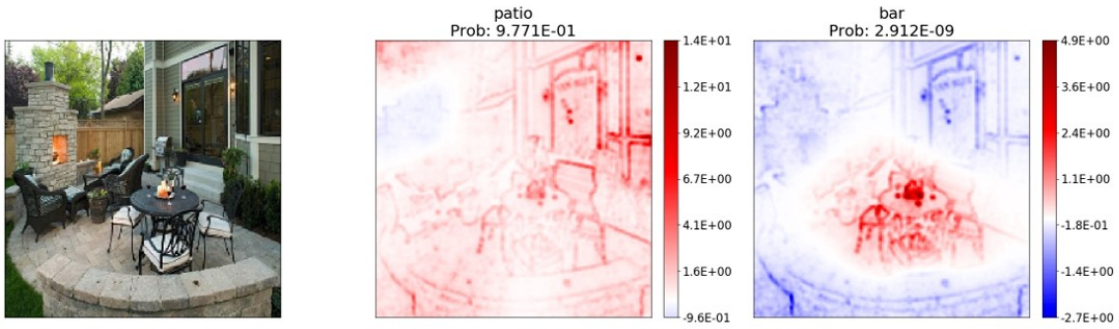


Figure 31: Patio evaluated for the maximum activation class (patio) and for the class: bar.

for the class "Physics laboratory" we can see in Figure 33 right most image that the result is the opposite. The objects on the ground are assigned positive contribution to the class while the frames are not considered typical objects of a physics laboratory.

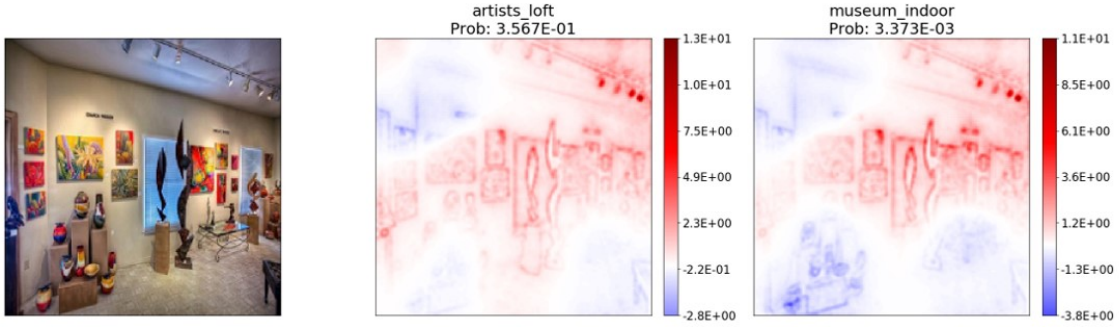


Figure 32: Art gallery evaluated for the maximum activation class (artist's loft) and for the class: museum indoor.

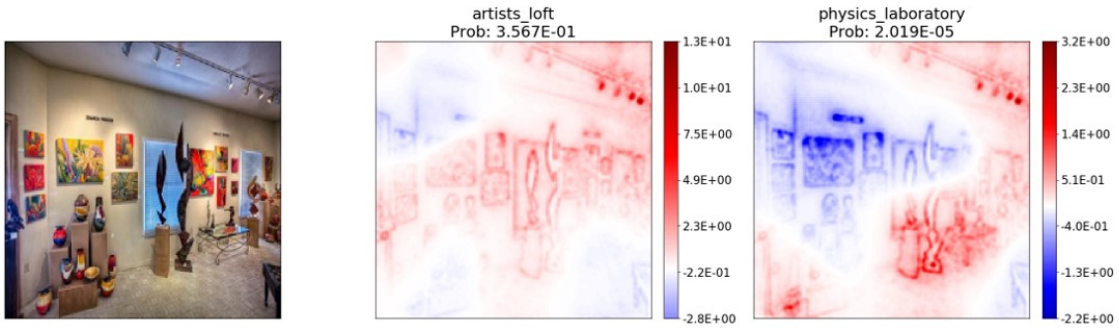


Figure 33: Art gallery evaluated for the maximum activation class (artist's loft) and for the class: physics laboratory.

The last image evaluated is a Kindergarden classroom. In Figure 34 we can see that for the correct class the kids are not considered typical of the class, the model is more focused on the objects of the background. If we evaluate the same image for the class restaurant kitchen we can see that the dishes and jars are considered typical of this the class, while all the background including toys

and book are not considered correct.

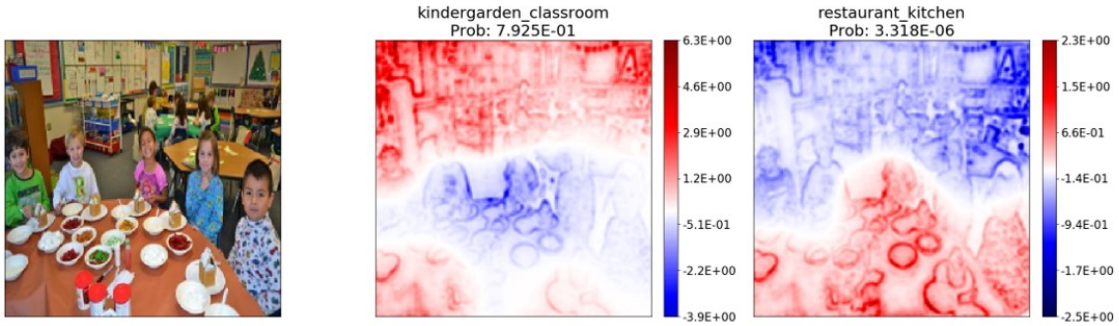


Figure 34: Kindergarden classroom evaluated for the maximum activation class (Kindergarden classroom) and for the class: restaurant kitchen.

6 Conclusions

In this thesis we have explored the field of algorithms that help a user to understand CNNs predictions. In the Section 2 we have presented different types of algorithms, all of them aimed at identifying the characteristics of an image that caused a specific prediction. After the initial review, we decided to focus the rest of the project on analyzing Layer-Wise Relevance Propagation.

In Section 3 we presented different rules that have been proposed to propagate the relevance and in Section 4 we observed their performance in a simplified set up: with a short CNN and the MNIST dataset. We analyzed in detail the values of the parameters inside the network, from activations to weights, and the relevances across layers to understand the difference that we saw in the saliency maps provided by the different rules and we propose three hypotheses to justify the effects observed. These hypotheses can be summarized as follow:

- The dense layers should use the ϵ-LRP rule to propagate the relevance.
- The convolutional layers should use a rule that does not use a negative sign that modifies the distribution of the saliency map obtained just after the dense layers.
- the Flat rule is a good option to propagate the relevance from the first convolutional layer to the input layer.

In Section 5 we have confirmed the three hypotheses in a more complex configuration, using a wider CNN with a VGG16 architecture and the Places 365 dataset. Then we combined all the hypotheses into a standard proposal for LRP methods which integrates all of them. The proposal consists on ϵ-LRP rule in the dense layers, $\alpha 1 \beta(-1)$-LRP rule in the convolutional layers and Flat rule in the input layer.

Finally, we have used this configuration to debug a trained model and we proved the capability of the method to give light to single predictions and help a human user to detect which features of the image were the most important to assign the label probability, and which features where against the class evaluated.

References

[1] BBVA. Bbva self and go tool. https://www.bbva.com/es/bbva-lanza-sistema-pagos-reconocimiento-facial/.

[2] Yilun Wang and Michal Kosinski. Deep neural networks are more accurate than humans at detecting sexual orientation from facial images. *Journal of Personality and Social Psychology*, 114:246–257, 02 2018.

[3] Klaus-Robert Müller Wojciech Samek Sebastian Lapuschkin, Alexander Binder. Understanding and comparing deep neural networks for age and gender classification., 08 2017. arXiv:1708.07689.

[4] Nature. Millions of black people affected by racial bias in health-care algorithms. https://www.nature.com/articles/d41586-019-03228-6.

[5] There's software used across the country to predict future criminals. and it's biased against blacks. n. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing.

[6] Carlos Guestrin Marco Tulio, Sameer Singh. "why should i trust you?" explaining the predictions of any classifier, 08 2016. arXiv:1602.04938.

[7] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society (Series B)*, 58:267–288, 1996.

[8] Su-In Lee Scott Lundberg. A unified approach to interpreting model predictions, 05 2017. arXiv:1705.07874.

[9] Montavon G Klauschen F Müller KR Samek W. Bach S, Binder A. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation, 07 2015. doi:10.1371/journal.pone.0130140.

[10] Alexander Binder Wojciech Samek Klaus-Robert Müller Grégoire Montavon, Sebastian Lapuschkin. Explaining nonlinear classification decisions with deep taylor decomposition, 05 2016. https://doi.org/10.1016/j.patcog.2016.11.008.

[11] Anshul Kundaje Avanti Shrikumar, Peyton Greenside. Learning important features through propagating activation differences, 04 2017. arXiv:1704.02685.

[12] Thomas Brox Martin Riedmiller Jost Tobias Springenberg, Alexey Dosovitskiy. Striving for simplicity: The all convolutional net, 12 2014. arXiv:1412.6806.

[13] Agata Lapedriza Aude Oliva Antonio Torralba Bolei Zhou, Aditya Khosla. Learning deep features for discriminative localization, 12 2015. arXiv:1512.04150.

[14] Abhishek Das Ramakrishna Vedantam Devi Parikh Dhruv Batra Ramprasaath R. Selvaraju, Michael Cogswell. Grad-cam: Visual explanations from deep networks via gradient-based localization, 10 2016. arXiv:1610.02391.

[15] Prantik Howlader Vineeth N Balasubramanian Aditya Chattopadhyay, Anirban Sarkar. Grad-cam++: Improved visual explanations for deep convolutional networks, 10 2017. arXiv:1710.11063.

[16] Taesup Moon Juyeon Heo, Sunghwan Joo. Fooling neural network interpretations via adversarial model manipulation, 02 2019. arXiv:1902.02041.

[17] Julius Adebayo Maximilian Alber Kristof T. Schütt Sven Dähne Dumitru Erhan Been Kim Pieter-Jan Kindermans†, Sara Hooker†. The (un)reliability of saliency methods, 02 2017. arXiv:1711.00867.

[18] Philipp Seegerer Miriam Hägele Kristof T. Schütt Grégoire Montavon Wojciech Samek Klaus-Robert Müller Sven Dähne Pieter-Jan Kindermans Maximilian Alber, Sebastian Lapuschkin. innvestigate neural networks!, 08 2018. arXiv:1808.04260.

[19] Tool to visualize cnn. http://alexlenail.me/nn-svg/alexnet.html.

[20] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[21] Grigorios Kalliatakis. Keras-vgg16-places365. https://github.com/GKalliatakis/Keras-VGG16-places365, 2017.